

IQI 04, Seminar 7

Produced with pdflatex and xfig

- Reversible classical computation.
- From irreversible to reversible computation.

E. "Manny" Knill: knill@boulder.nist.gov

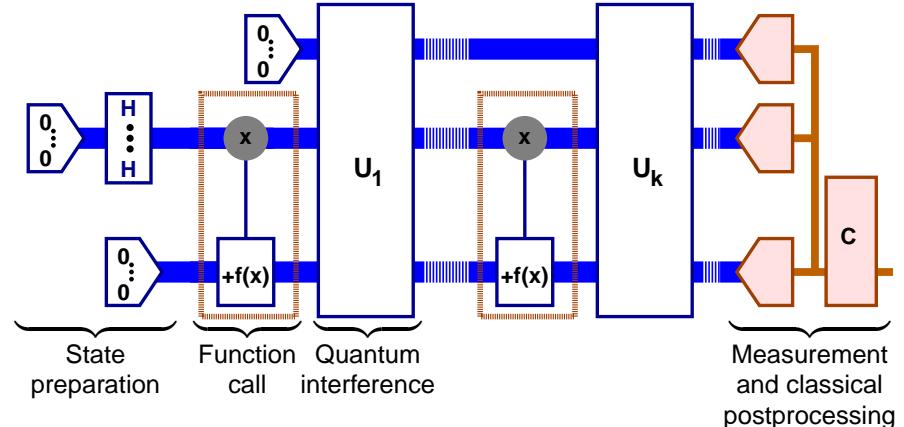


TOC

2

Recognizing Patterns in Functions

- Algorithm structure for determining some **Property**(f).
 - Multiple function calls:

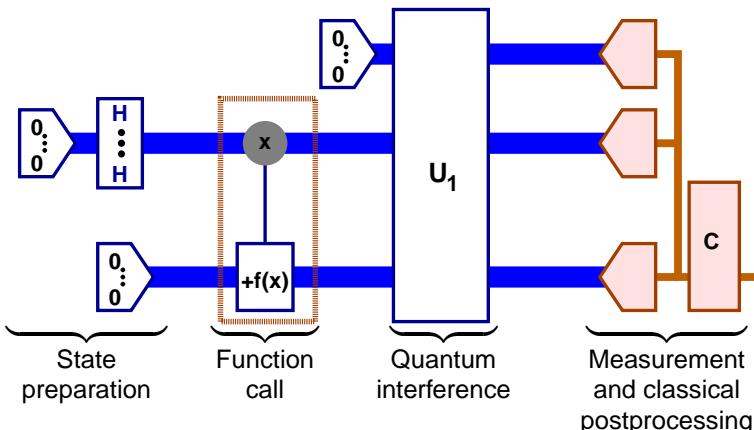


Measurement and classical postprocessing

TOC

Recognizing Patterns in Functions

- Algorithm structure for determining some **Property**(f).
 - One function call:

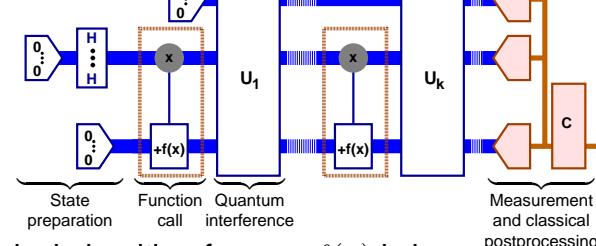


TOC

1

Recognizing Patterns in Functions

- Algorithm structure for determining some **Property**(f).
 - One function call:



Measurement and classical postprocessing

- A classical algorithm for $x \mapsto f(x)$ is known.
 - How to implement $\sum_x \alpha_x |xy\rangle \rightarrow \sum_x \alpha_x |x(y \oplus f(x))\rangle$?
- Solution:
 1. Algorithm \rightarrow circuit($|x|$).
 2. Irreversible gates \rightarrow reversible gates and memory.
 3. Reversibly erase memory.
 4. Bits \rightarrow qubits. Reversible gates \rightarrow unitary gates.

TOC

3

Example: Number Comparison

- Algorithm for comparing two binary numbers.

$\text{COMP}(x, y)$

Input: Zero-filled n -bit numbers $x = x_{n-1} \dots x_0$ and $y = y_{n-1} \dots y_0$.

Output: 0 if $x < y$ and 1 if $x \geq y$.

```

 $k \leftarrow n - 1$ 
 $c \leftarrow -1$ 
while  $k \geq 0 \& c < 0$ 
  if  $x_k < y_k$ 
     $c \leftarrow 0$ 
  else if  $x_k > y_k$ 
     $c \leftarrow 1$ 
  end
   $k \leftarrow k - 1$ 
end
if  $c < 0$  then  $c \leftarrow 1$ 
return  $c$ 

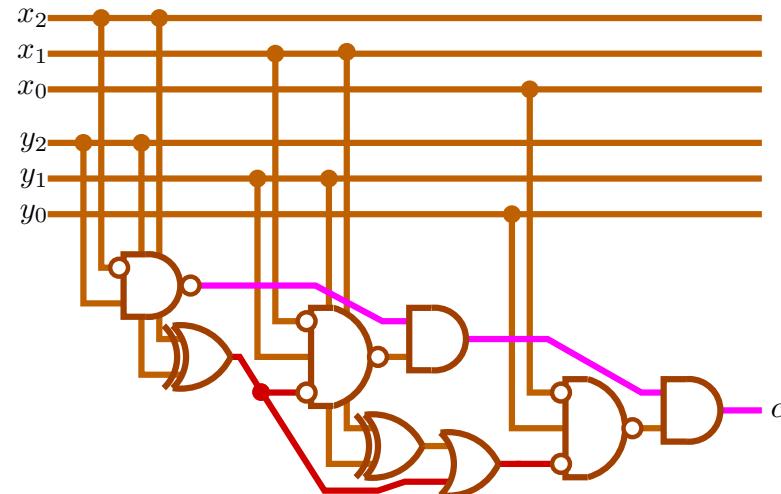
```

$k : 1$	$x : 0 \ 1 \ 0 \ \boxed{1} \ 0 \ 1$
$c : 1$	$y : 0 \ 1 \ 0 \ \boxed{0} \ 1 \ 1$

4
TOC

Classical Comparison Circuit

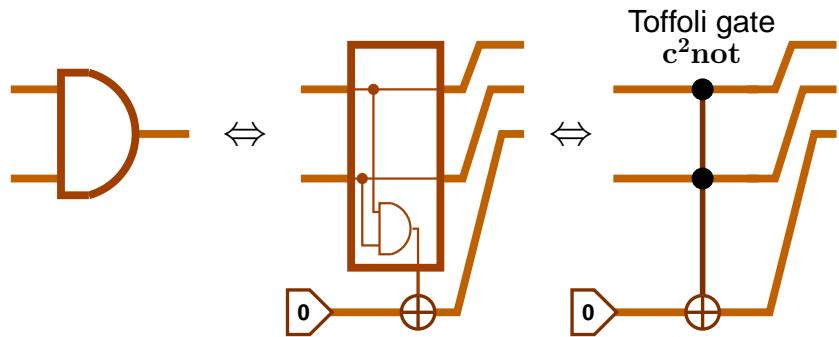
- 3 bit comparison circuit.



5
TOC

Converting to Reversible Logic

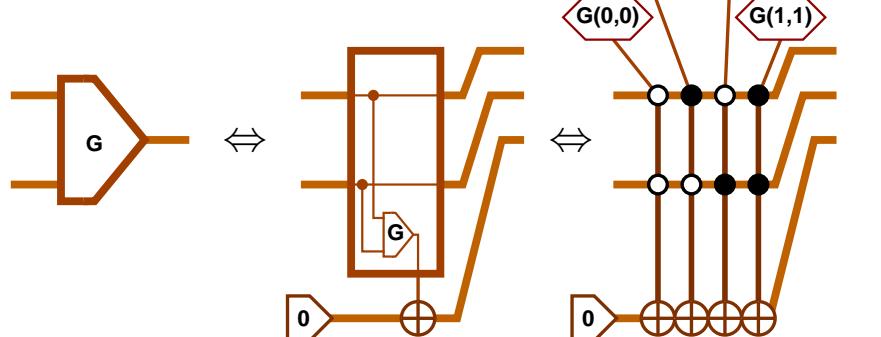
- Reversing the and gate.



6
TOC

Converting to Reversible Logic

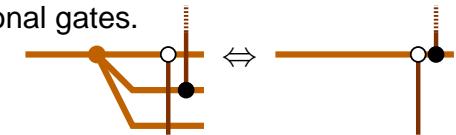
- Reversing a general gate.



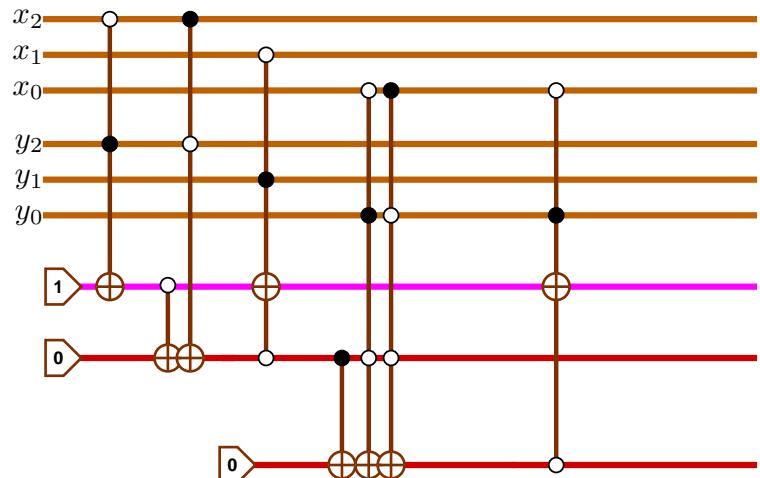
7
TOC

- ... then optimize the conditional gates.

- Remove redundant fanout.



Reversifying the Comparison Circuit

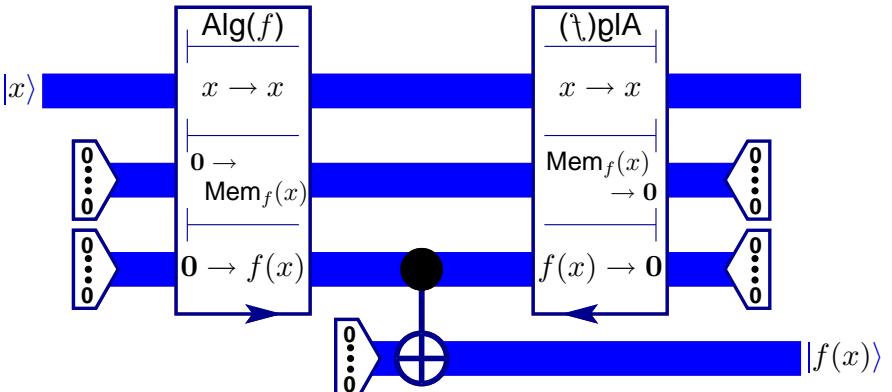


8

TOC

Erasing Memory

- Erasing memory by copying output and reversing.

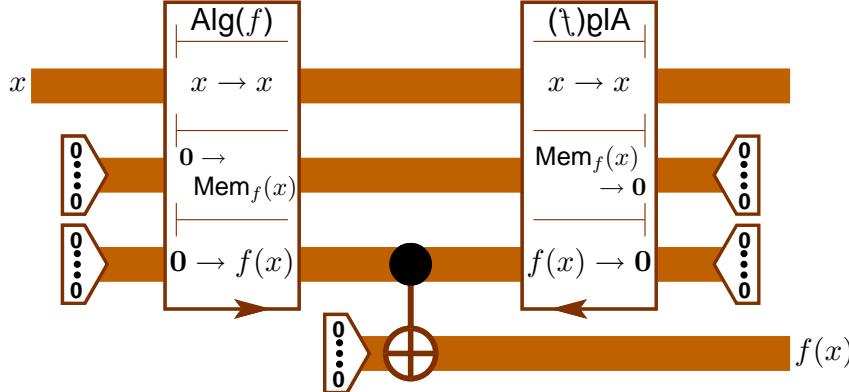


10

TOC

Erasing Memory

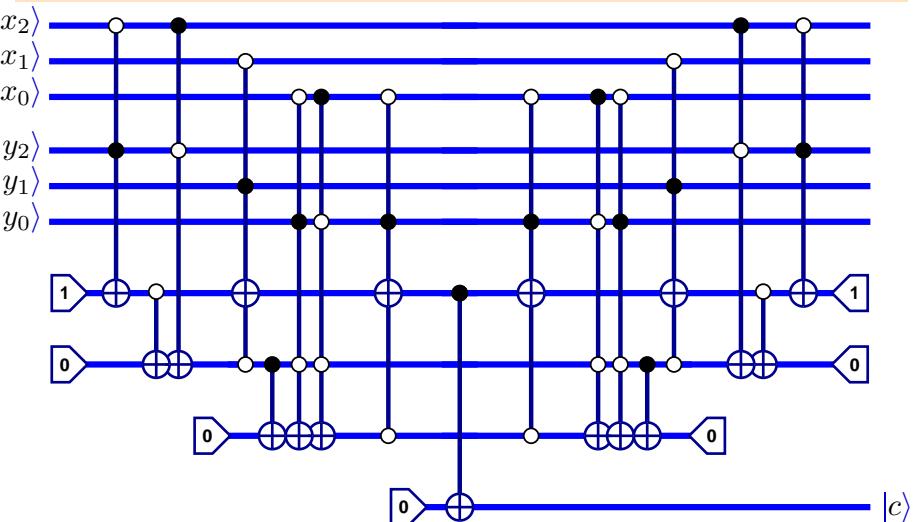
- Erasing memory by copying output and reversing.



9

TOC

Coherent Comparison

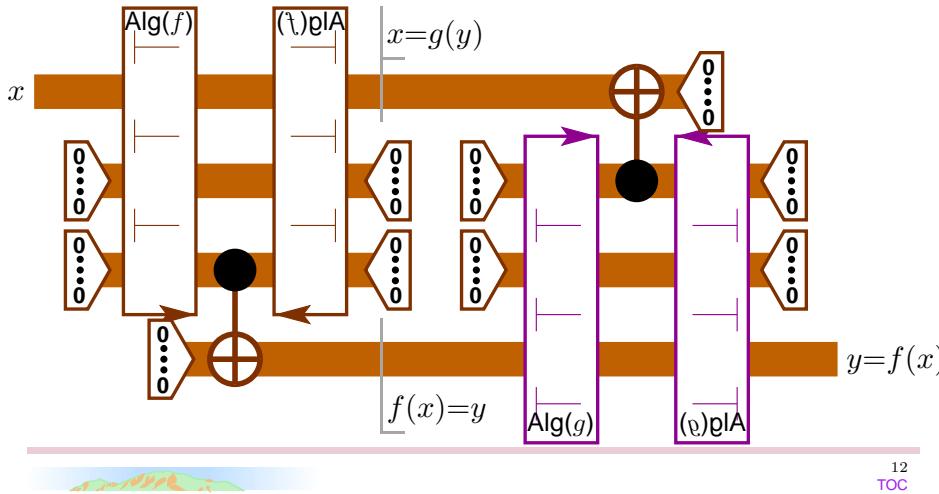


11

TOC

Invertible Functions

- When can one coherently implement $\sum_x \alpha_x |x\rangle \rightarrow \sum_x \alpha_x |f(x)\rangle$?
 - If there exists g such that $g(f(x)) = x$ (f is invertible).
- Network implementation given algorithms for f and $g = f^{-1}$.



References

- [1] C. H. Bennett, G. Brassard, S. Breidbart, and S. Wiesner. Quantum cryptography, or unforgeable subway tokens. In *Advances in Cryptology: Proceedings of Crypto'82*, pages 267–275. Plenum Press, 1982.
[2] C. H. Bennett. Time/space trade-offs for reversible computation. *SIAM J. Comput.*, 18:766–776, 1989.

14
TOC

Contents

Title: IQI 04, Seminar 7	0	Reversing the Comparison Circuit	8
Recognizing Patterns in Functions I	1	Erasing Memory I	9
Recognizing Patterns in Functions II	2	Erasing Memory II	10
Recognizing Patterns in Functions III	3	Coherent Comparison	11
Example: Number Comparison	4	Invertible Functions	12
Classical Comparison Circuit	5	References	14
Converting to Reversible Logic I	6		
Converting to Reversible Logic II	7		

13
TOC